**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

**по дисциплине «Введение в нереляционные базы данных»**

Тема: Информационный сайт об астероидах и

метеоритах на основе NASA API

|  |  |  |
| --- | --- | --- |
| Студент гр. 8303 |  | Почаев Н.А. |
| Студент гр. 8303 |  | Перелыгин Д.С. |
| Студент гр. 8304 |  | Алтухов А.Д. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2021

**ЗАДАНИЕ**

**на ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

|  |  |  |
| --- | --- | --- |
| Студенты  Почаев Н.А.  Перелыгин Д.С.  Алтухов А.Д. | | |
| Группа 8303 и 8304 | | |
| Тема работы: Информационный сайт об астероидах и метеоритах на основе NASA API | | |
| Исходные данные:  Создание веб-сайта, предоставляющего функциональность по просмотру актуальных данных по метеоритам и астероидам, предоставляемых NASA API. Данные представляются в виде наглядных карт с отметкой мест падения, таблиц с полной известной информацией о небесном теле. Также по предоставленным данных возможен просмотр статистики в виде графиков. Имеется функционал добавления и импорта / экспорта данных из БД. | | |
| Содержание пояснительной записки:  «Содержание»  «Введение»  «Сценарии использования»  «Модель данных»  «Разработка веб-сайта»  «Выводы»  «Приложение» | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания: 01.09.2020 | | |
| Дата сдачи реферата: | | |
| Дата защиты реферата: | | |
| Студент гр. 8303 |  | Почаев Н.А. |
| Студент гр. 8303 |  | Перелыгин Д.С. |
| Студент гр. 8304 |  | Алтухов А.Д. |
| Преподаватель |  | Заславский М.М. |

**Аннотация**

В рамках указанной дисциплины предполагается создание веб-сайта и (или) приложения в команде с использованием технологии NoSQL. В виде предметной области данного проекта выбрано использование NASA API для предоставления информации об астероидах и метеоритах.

Найти исходный код и дополнительную информацию можно по ссылке: https://github.com/moevm/nosql2h21-nasa

**Summary**

In the framework of the mentioned discipline, it is supposed to create a website and (or) an application in a team using NoSQL technology. The subject area of this project is the use of NASA API to provide information about asteroids and meteorites.

Find the source code and more information at this link:

https://github.com/moevm/nosql2h21-nasa

**содержание**

|  |  |
| --- | --- |
| [1. ВВЕДЕНИЕ 6](#_Toc52402883)  [2. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ 7](#_Toc52402884)  [3. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ 8](#_Toc52402885)  [4. МОДЕЛЬ ДАННЫХ 2](#_Toc52402886)14  [6. ЗАКЛЮЧЕНИЕ 1](#_Toc52402888)22 |  |

# введение

Целью данной работы является создание веб-сайта, предоставляющего функциональность по получению визуализированной информации (карты, графики, таблицы) по астероидам и метеоритам, известным NASA. Также данный сайт предоставляет функциональность по просмотру текущего положения МКС. Поддержано добавление новых данных, а также импорт и экспорт базы данных. Использованный стек: Node.js, Handlebars, MongoDB, Docker.

**1.** Качественные требования к решению

Требуется разработать user-friendly веб-сайт, реализующий и предоставляющий пользователю все заявленные функции. Представление данных должно быть наглядным и интерактивным. В качестве системы управления базами данных используется MongoDB.**2.** сценарии использования

**2.1.** UI реализованного сайта

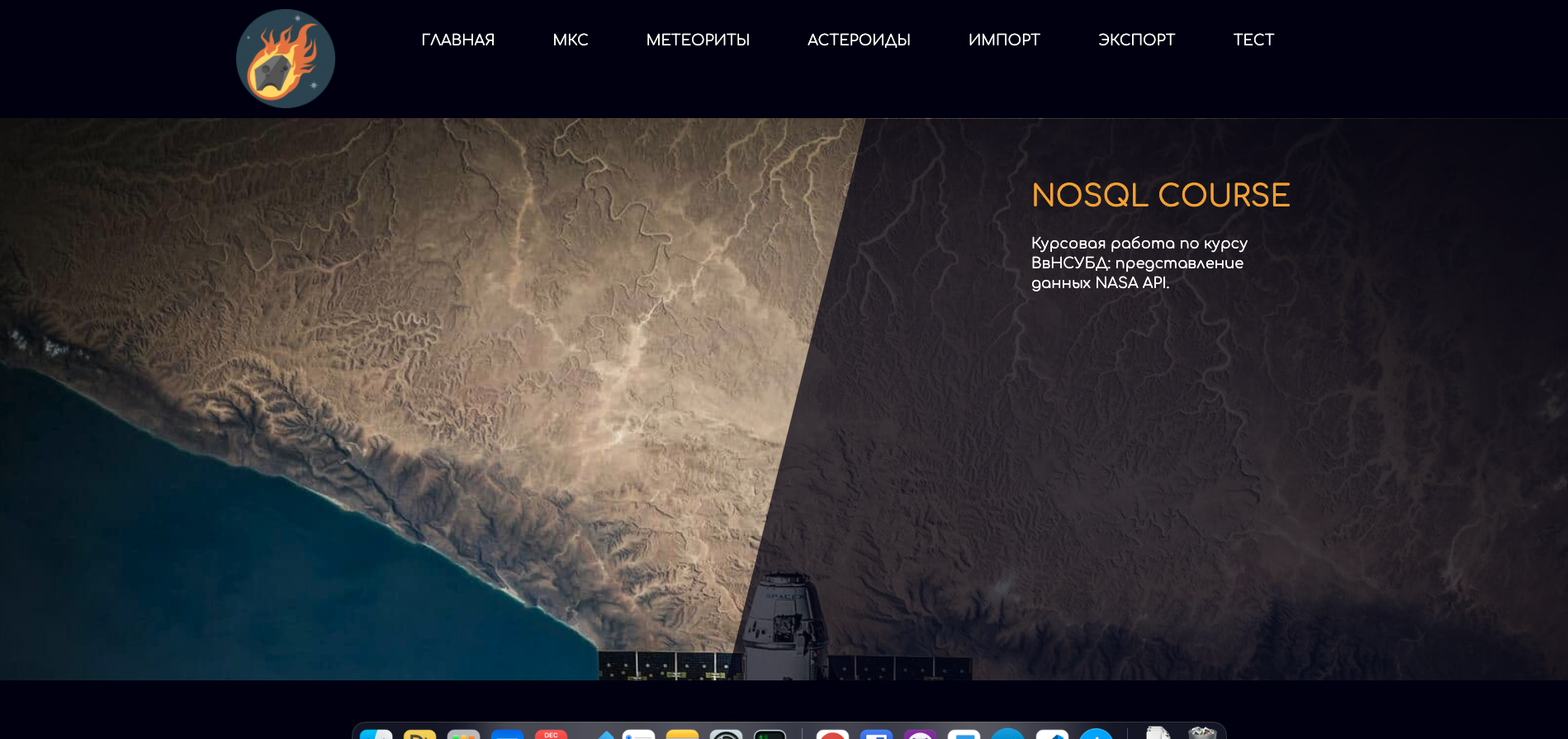
****

Рисунок 1 – Стартовая станица сайта

****

Рисунок 2 – Карта метеоритов с возможностью фильтрации

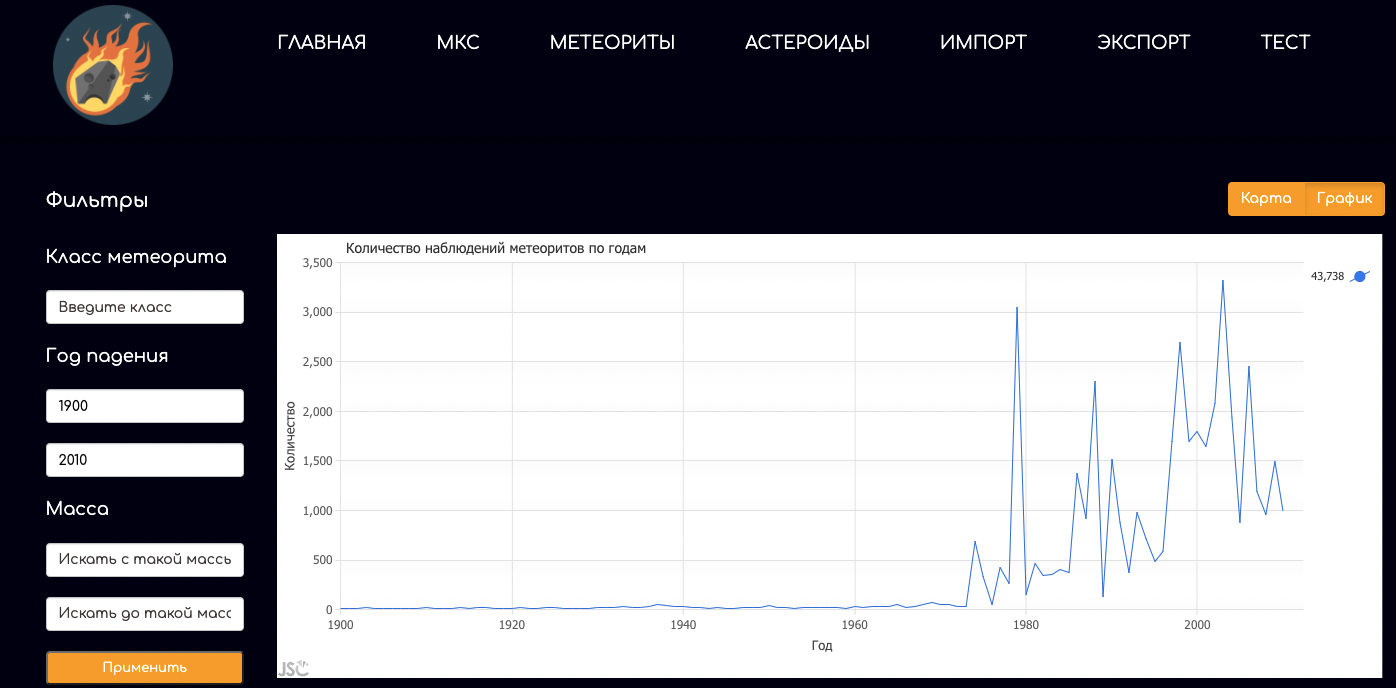
****

Рисунок 3 – График падения метеоритов по годам

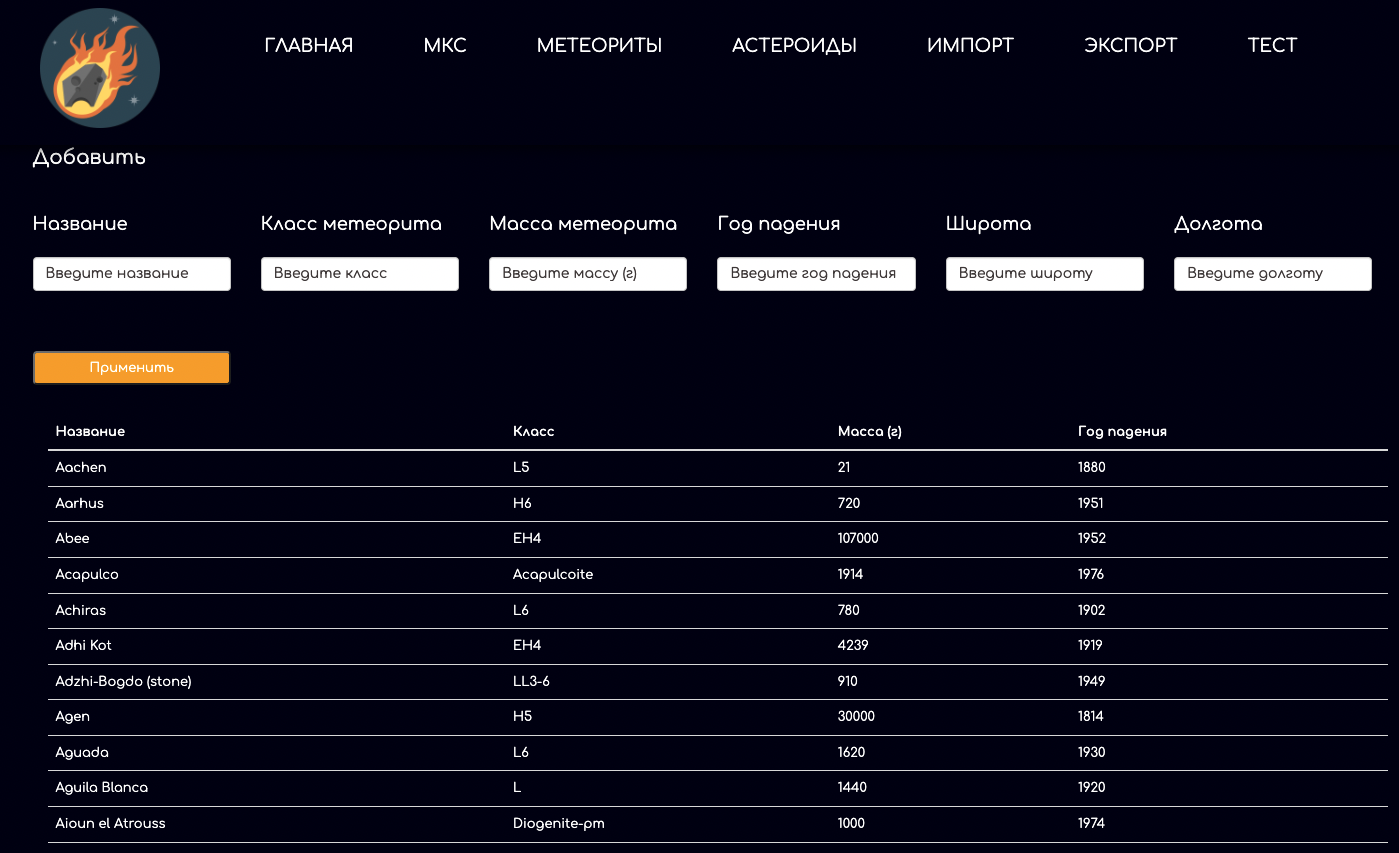
****

Рисунок 4 – Таблица всех метеоритов с возможностью добавления новых

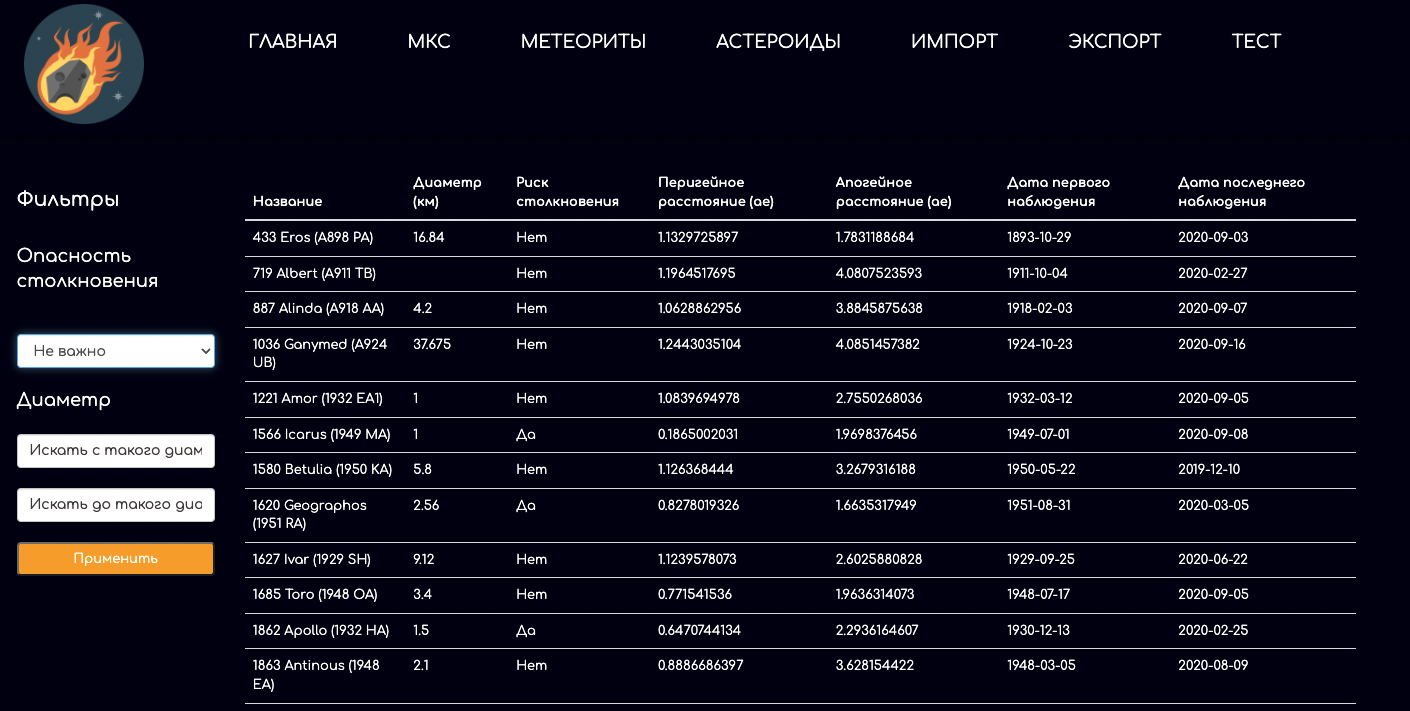
****

Рисунок 5 – Таблица астероидов с полной информацией и возможностью фильтрации

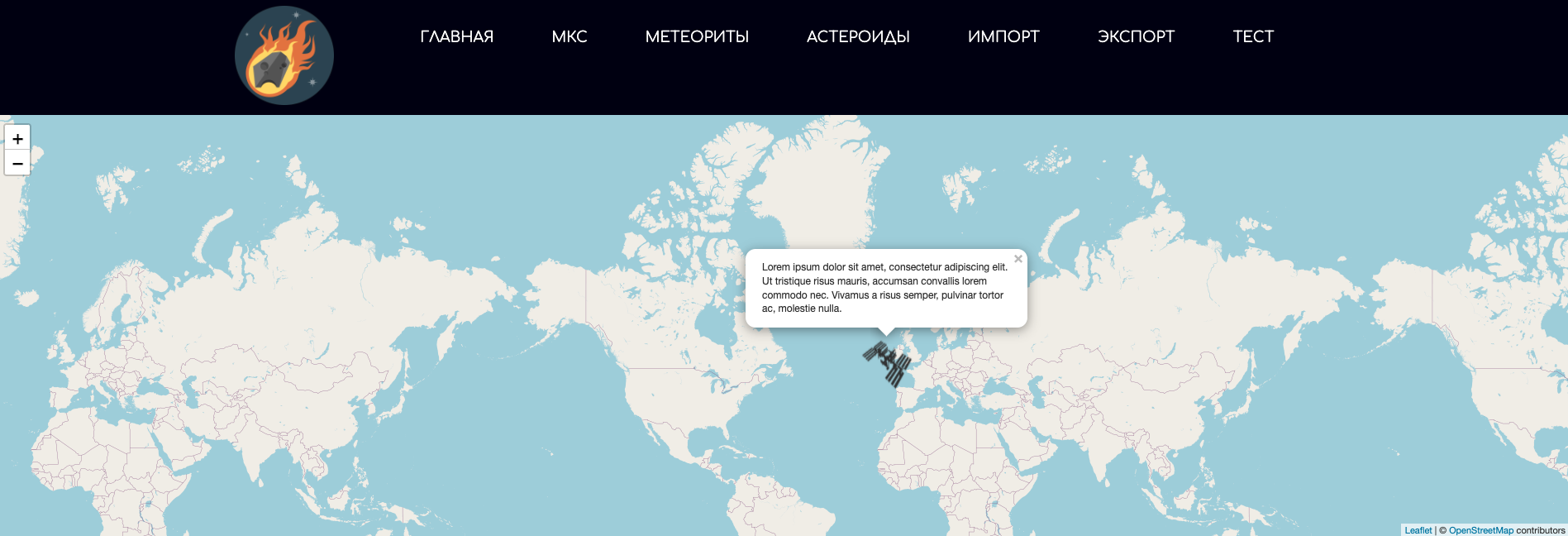
****

Рисунок 6 – Карта текущего местоположения МКС

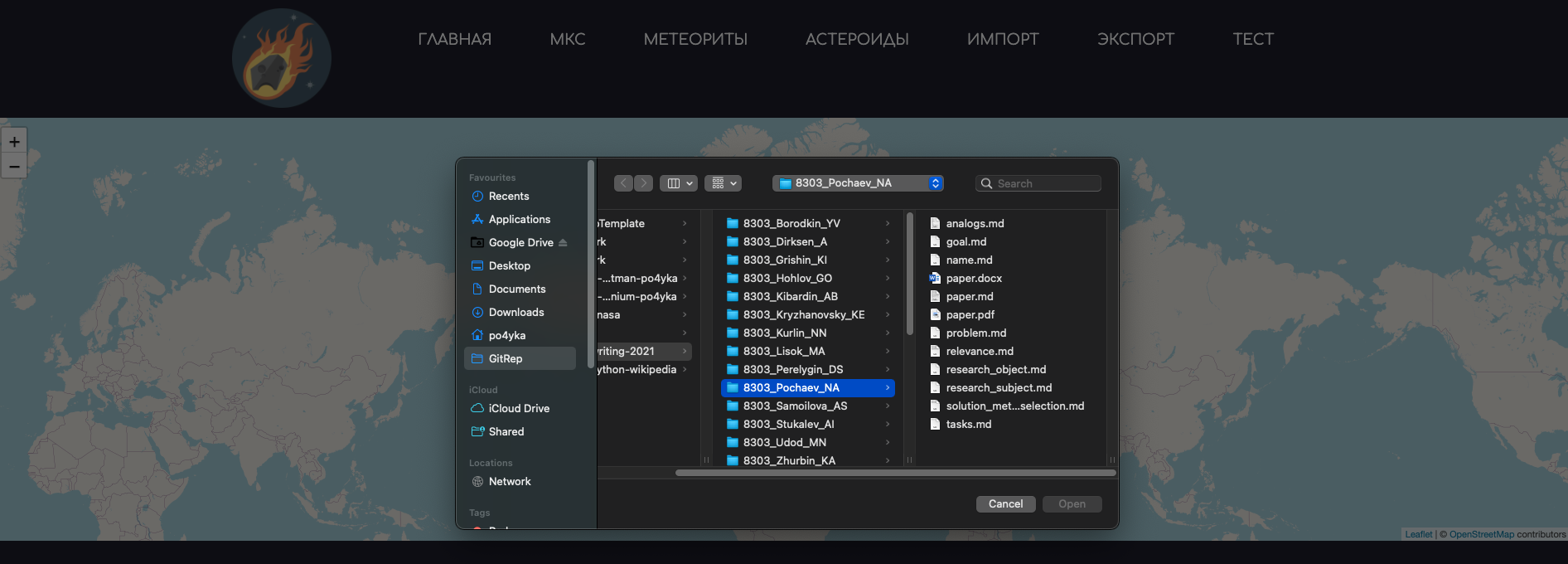
****

Рисунок 7 – Окно экспорта и импорта БД

**2.2.** Сценарии использования

1. **Сценарий использования - "Просмотр информации о функционале проекта"**

*Действующее лицо:* Пользователь

*Основной сценарий:*

* 1. Выполнить переход на главную страницу нажатием на кнопку "Главная" на верхней панели.
  2. Пользователь видит описание доступных функций.

1. **Сценарий использования - "Просмотр информации о текущем положении МКС"**

*Действующее лицо:* Пользователь

*Основной сценарий:*

1. Выполнить переход на страницу МКС нажатием на кнопку "МКС" на верхней панели.
2. Пользователь получает возможность интерактивного взаимодействия с картой.
3. Пользователь видит иконку, обозначающую МКС, на карте.

*Альтернативный сценарий:*

1. API недоступно – невозможно получить информацию о МКС.
2. **Сценарий использования - "Просмотр видеопотока с МКС"**

*Действующее лицо:* Пользователь

*Основной сценарий:*

* 1. Выполнить переход на страницу МКС нажатием на кнопку "МКС" на верхней панели.
  2. Нажать кнопку воспроизведения на видео-плеере.

*Альтернативный сценарий:*

1. API недоступно – невозможно получить информацию о МКС.
2. Видеопоток недоступен из-за потери сигнала или использования сети связи для нужд станции.
3. **Сценарий использования - "Просмотр информации о метеоритах"**

*Действующее лицо:* Пользователь

*Основной сценарий:*

* 1. Выполнить переход на страницу метеоритов нажатием на кнопку "Метеориты" на верхней панели.
  2. Пользователь получает доступ к таблице со списком метеоритов.
  3. Пользователь меняет отображаемую таблицу по своему усмотрению с помощью фильтрации (см. use-case "Фильтрация списка метеоритов").
  4. Пользователь получает доступ к интерактивной карте с отмеченными метеоритами.
  5. Пользователь получает к доступ к маркеру на карте путем нажатия на объект в таблице.
  6. Пользователь получает доступ к информации о маркере после нажатия на него.
  7. Пользователь получает доступ к сортировке столбцов таблицы.
  8. Пользователь нажимает на кнопку "График" и интерактивная карта заменяется графиком, показывающим зависимость количества упавших метеоритов от года.

*Альтернативный сценарий:*

1. Пользователь нажимает на кнопку "Карта" и возвращается к интерактивной карте.
2. По умолчанию фильтрация отсутствует – показываются все метеориты.
3. API недоступно – невозможно получить информацию о метеоритах.
4. **Сценарий использования - "Фильтрация списка метеоритов"**

*Действующее лицо:* Пользователь

*Предусловие:* Нахождение на странице "Метеориты"

*Основной сценарий:*

* 1. Пользователь заполняет поле "Класс метеорита" – доступные классы можно узнать из таблицы со списком всех метеоритов.
  2. Пользователь заполняет поле "Год падения: от" – заполняется целым числом.
  3. Пользователь заполняет поле "Год падения: до" – заполняется целым числом.
  4. Пользователь заполняет поле "Масса: от" – масса измеряется в граммах, заполняется произвольным положительным числом.
  5. Пользователь заполняет поле "Масса: до" – масса измеряется в граммах, заполняется произвольным положительным числом.
  6. Пользователь нажимает кнопку "Применить" и таблица обновляется соответственно настроенным фильтрам.

*Альтернативный сценарий:*

1. Пользователь не заполняет поле – фильтры игнорируют этот параметр.
2. **Сценарий использования - "Просмотр информации об астероидах"**

*Действующее лицо:* Пользователь

*Основной сценарий:*

* 1. Выполнить переход на страницу астероидов нажатием на кнопку "Астероиды" на верхней панели.
  2. Пользователь получает доступ к полному списку астероидов, находящихся рядом с Землей.
  3. Пользователь меняет отображаемую таблицу по своему усмотрению с помощью фильтрации (см. use-case "Фильтрация списка астероидов").
  4. Пользователь получает доступ к сортировке столбцов таблицы.

*Альтернативный сценарий:*

1. По умолчанию фильтрация отсутствует – показываются все астероиды.
2. API недоступно – невозможно получить информацию об астероидах.
3. **Сценарий использования - "Фильтрация списка астероидов"**

*Действующее лицо:* Пользователь

*Предусловие:* Нахождение на странице "Астероиды"

*Основной сценарий:*

* 1. Пользователь выбирает значение для поля "Опасность столкновения" – доступные значения: "Не важно", "Да", "Нет".
  2. Пользователь заполняет поле "Диаметр: от" – диаметр измеряется в километрах, заполняется произвольным положительным числом.
  3. Пользователь заполняет поле "Диаметр: до" – диаметр измеряется в километрах, заполняется произвольным положительным числом.
  4. Пользователь нажимает кнопку "Применить" и таблица обновляется соответственно настроенным фильтрам.

*Альтернативный сценарий:*

1. Пользователь не заполняет поле – фильтры игнорируют этот параметр.

**3.** Модель данных

**3.1.** Схема данных в SQL

Схема данных, как она бы выглядела при использовании стандартной реляционной БД приведена ниже на рис. 8.

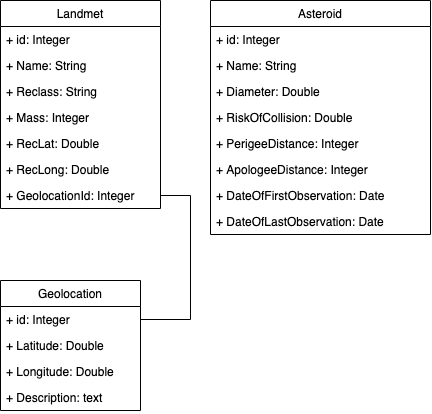


Рисунок 8 – Схема данных в SQL

**3.2.** Схема данных в NoSQL, сравнение с SQL

Схема данных в NoSQL БД (в контексте данной работы – MongoDB) представлена на рис. 9. Представление в формате JSON представлены на рис. 10-11.

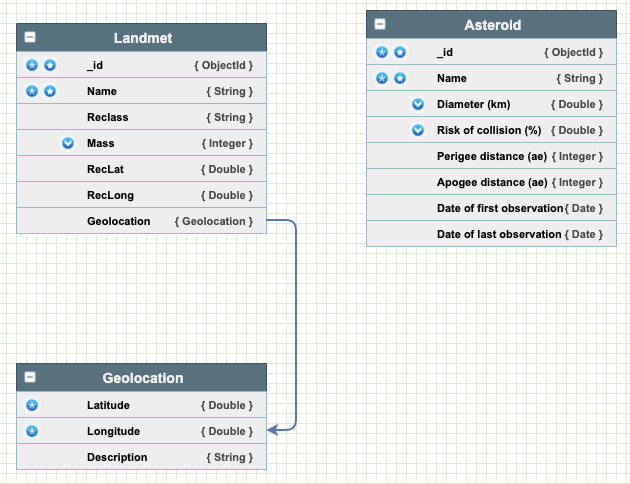


Рисунок 9 – Схема данных в NoSQL

* Landmet:
  + id - уникальный идентификатор;
  + Name - название / имя метеорита;
  + Reclass (class) - класс метеорита;
  + Mass - масса;
  + RecLat - широна нахождения;
  + RecLong - долгота нахождения;
  + GeolocationId - локация падения с широтой и долготой.
* Asteroid:
  + id - уникальный идентификатор;
  + Diameter - диаметр;
  + RiskOfCollision - риск столкновения;
  + PerigeeDistance - расстояние Перегилия;
  + ApologeeDistance - апогейное расстояние;
  + DateOfFirstObservation - дата первого наблюдения;
  + DateOfLastObservation - дата последнего наблюдения.
* Geolocation:
  + id - уникальный идентификатор;
  + Latitude - ширина падения;
  + Longitude - долгота падения;
  + Description - описание падения.

****

Рисунок 10 – Схема данных метеорита в MongoDB



Рисунок 11 – Схема данных астероида в MongoDB

Каждая сущность, как астероид, так и метеорит, представляют собой отдельные документы. Поле Geolocation у сущности Landmet является вложенным документом.

**3.3.** Список сущностей

В данном проекте осуществяется хранение и обработка данных для двух явлений: метеоритов и астероидов.

* Метеорит - крупные метеорные тела, которые падают на Землю.
* Астероид - малые планеты - твёрдые небесные тела, которые значительно меньше планет по размеру. Движутся по орбитам.

**3.4.** Оценка объёма информации

Рассмотрим сущность Landmet для наглядности (наличие вложенного документа). Пусть совокупность полей Name, Reclass, Mass, RecLat, RecLong занимает N памяти, M - количество упавших метеоритов, а вложенный домент Geolocation (поля Latitude, Longitude, Description) - K.

Тогда для хранения данных будет занято:

* MongoDB: M \* (N + K) = M \* N + M \* K
* SQL: M \* N + \* M \* K

Тогда, согласно [документации](https://bsonspec.org/spec.html), N = 32 байт название (строка в UTF-8) + 64 байт масса + 32 байт класс + 32 \* 2 (координаты) = 160 байт. В реляционной БД результаты будут аналогичными. Таким образом, кол-во чистой памяти, занимаемой реляционной БД, не отличается от MongoDB.

Количество памяти, занимаемой MongoDB с учётом типа <ObjectId> (12 байт) и overhead (e\_list + zero-termination) в 1 байт для каждого элемента = 12 + 5 + 160 = 177 байт.

Количество памяти, занимаемой реляционной БД с учётом полей id (4 байта): 4 байта \* 4 id \* 2 (с учётом ссылок) + 160 = 192 байт.

Таким образом, MongoDB выигрывает реляционной БД при учёте всей информации, хранимой в моделе (расчёты для одной запииси / документа).

Формула роста памяти для MongoDB при увеличении кол-ва моделей будет иметь вид: N \* (12 + 5 + 160) = 177N, а для SQL: 192N.

Как следствие, использования MongoDB по памяти в данном случай будет эффективнее на 177/192.

**3.5.** Запросы

* Запрос на поиск метеорита по id по всей коллекции (без учёта индексов) по местоположению:
  + MongoDB: O(M)
  + SQL DB: O(2M^2), т.к. необходимо сделать 2 раза операцию JOIN на таблицу Geoposition.
* Запросы на текущий момент осуществляются стандартным путём. (WIP)

**3.6.** Исследование итоговой производительности

В ходе экспериментов с готовой системой измерены следующие характеристики работы приложения:

* время обработки запросов к БД;
* фактический объем хранимых данных.

Результаты измерений представлены в табл. 1-2 соответственно. И на графиках далее.

**Таблица 1 – Зависимость кол-ва записей в БД и времени запроса**

|  |  |
| --- | --- |
| **Кол-во записей** | **Время, мс** |
| 2285 | 3,212599999 |
| 4570 | 5,19884 |
| 6855 | 6,370085 |
| 9170 | 8,119725 |
| 11455 | 9,66861 |
| 13740 | 10,097995 |
| 16025 | 12,52824 |
| 18310 | 13,633155 |
| 20595 | 13,9511 |
| 22880 | 13,999335 |
| 25165 | 15,42216 |
| 27450 | 17,568295 |
| 29735 | 18,37964 |
| 32020 | 21,001875 |
| 34405 | 22,514435 |
| 36590 | 23,777655 |
| 38875 | 25,0248 |
| 41160 | 28,460135 |
| 43445 | 30,5348 |
| 45730 | 31,76351 |

**Таблица 2 – Зависимость кол-ва записей в БД и времени запроса**

|  |  |
| --- | --- |
| **Кол-во записей** | **Объём памяти, байт** |
| 2285 | 469377 |
| 4570 | 941020 |
| 6855 | 1410916 |
| 9170 | 1874556 |
| 11455 | 2377132 |
| 13740 | 2866904 |
| 16025 | 3371357 |
| 18310 | 3841239 |
| 20595 | 4291188 |
| 22880 | 4758899 |
| 25165 | 5229756 |
| 27450 | 5673780 |
| 29735 | 6103881 |
| 32020 | 6577215 |
| 34405 | 7065333 |
| 36590 | 7540538 |
| 38875 | 8011883 |
| 41160 | 8486672 |
| 43445 | 8961453 |
| 45730 | 9400795 |

Рисунок 12 – Зависимость времени выполнения запроса к БД от кол-ва записей

Рисунок 13 – Зависимость потребляемой памяти БД от кол-ва записей

Как из приведённых практических данных, спроектированная и реализованная БД, использованная в созданном веб-сайте, полностью соответствует заявленным положениям о теоретической работе по потребляемой памяти и скорости выполнения запросов – сложность и рост являются линейными.

# заключение

В результате выполнения данной работы была спроектирована и разработана БД NoSQL с использованием MongoDB, полностью соответствующая своей области применения и производительная, согласно теоретическим подсчётам. Также был разработан user-friends сайт, позволяющий пользователю видеть подробную структурированную информацию о метеоритах и астероидах, предоставляемую NASA.